UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/992,736	11/13/2001	Rajeev Bharadhwaj	PA2314US	1539

22830 7590 04/21/2004

CARR & FERRELL LLP
2200 GENG ROAD
PALO ALTO, CA 94303

EXAMINER	
BULLOCK JR, LEWIS ALEXANDER	
ART UNIT	**PAPER NUMBER**
2126	3

DATE MAILED: 04/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/992,736	BHARADHWAJ, RAJEEV	
	Examiner	Art Unit	
	Lewis A. Bullock, Jr.	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 November 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 November 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. New corrected drawings are required in this application because of Draftperson's Review. Applicant is advised to employ the services of a competent patent draftsperson outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

Priority

An application in which the benefits of an earlier application are desired must contain a specific reference to the prior application(s) in the first sentence of the specification of in an application data sheet (37 CFR 1.78(a)(2) and (a)(5)). The specific reference to any prior nonprovisional application must include the relationship (i.e., continuation, divisional, or continuation-in-part) between the applications except when the reference is to a prior application of a CPA assigned the same application number.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double

patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1-35 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-27 of U.S. Patent No. 6,360,244. Although the conflicting claims are not identical, they are not patentably distinct from each other because the patent teaches the protecting the domain code from the user code at a protection level by context switching between the user process context and the domain process context wherein the user context process has a non-executable reserve portion storing the domain code or by the context switching establishing two levels of protection within the protection level.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over KARGER (U.S. Patent 5,210,874) in view of "Software Fault Tolerance in Architectures with Hierarchical Protection Levels" by OZAKI.

As to claim 1, KARGER teaches a computer implemented method for multi-level memory domain protection by protecting the domain code (called domain code)

executing at the second protection level from the user code (calling domain code) executing at the second protection level by context switching between the user process context (calling domain context) and the domain process context (called domain context) (col. 4, line 36 – col. 6, line 34). However, KARGER does not explicitly mention that the domain process context and the user process context have respective operating system code executing at the first protection level.

OZAKI teaches a method for multi-level memory domain protection, comprising the steps of: establishing a domain process context (process T1) having operating system code (MB1), executing at a first protection level (privilege level 1/ privilege level 0), and domain code (application code), executing at a second protection level (privilege level 3); and establishing a user process context (process T2) having the operating system code (MB2), executing at the first protection level (privilege level 1 / privilege level 0), and user code (application code / T2_Proc) executing at the second protection level (privilege level 3) such that one process invokes the other (pg. 39, Time-outs; Table 1). Therefore, it would be obvious to one skilled in the art to combine the teachings of KARGER with the teachings of OZAKI in order to facilitate system protection during inter-process communication (pg. 39).

As to claim 10, KARGER teaches a computer implemented method for multi-level memory domain protection by inter-group context switching between the user process context (calling domain context) and the domain process context (called domain context) (col. 4, line 36 – col. 6, line 34). However, KARGER does not teach the context

switching entails steps of executing calling code, branching to a call gate, executing linking code, branching to a target code, and executing the second process.

OZAKI teaches a computer-implemented method for multi-level memory domain protection, comprising the steps of: executing calling-code (selector call for T2_Proc) in a first process pair (T1) calling for execution of targeted code in a second process pair (T2); branching to a call gate target code-segment (call gate) in the calling-code corresponding to the targeted code; executing linking-code in the target code-segment and entering operating system code (MB1) in the first process pair; branching to and executing the targeted code (T2_Proc complete successfully); and executing second process pair to first process pair return code (return parameters to T1) (pg. 39). Therefore, it would be obvious to one skilled in the art to combine the teachings of KARGER with the teachings of OZAKI in order to facilitate system protection during inter-process communication (pg. 39).

As to claim 20, KARGER teaches a system for multi-level memory domain protection by protecting the domain code (called domain code) executing at the second protection level from the user code (calling domain code) executing at the second protection level by intra-group context switching between the user process context (calling domain context) and the domain process context (called domain context) (col. 4, line 36 – col. 6, line 34). However, KARGER does not explicitly mention that the domain process context and the user process context have respective operating system code executing at the first protection level.

OZAKI teaches a system for multi-level memory domain protection, the system comprising: a user process (process T2), for executing operating system code (MB2) at a first protection level (privilege level zero or one) and for executing user code (application code of process T2) at a second protection level (privilege level 3); and a domain process (process T1), for executing the operating system code (MB1) at the first protection level (privilege level zero or one), for executing domain code (application code of process T1) at the second protection level (privilege level 3) such that one process invokes the other (pg. 39). Refer to claim 1 for the motivation to combine.

As to claim 2, OZAKI teaches the domain code (application code for process T1) includes domain-to-user control transfer instructions (selector call for T2_Proc) (pg. 39). It would be obvious that since the processes call one another through the mailboxes that communication is made by transferring through privilege level zero or one where the operating system resides in order to switch contexts as disclosed in KARGER.

As to claim 3, OZAKI teaches the user code (application code of process T2) includes user-to-domain control transfer instructions (T2 passing returned parameters to T1) (pg. 39). It would be obvious that since the processes call one another through the mailboxes that communication is made by transferring through privilege level zero or one where the operating system resides in order to switch contexts as disclosed in KARGER.

As to claim 6, KARGER teaches a computer implemented method for multi-level memory domain protection by intra-group context switching between the user process context (calling domain context) and the domain process context (called domain context) (col. 4, line 36 – col. 6, line 34). However, KARGER does not teach the context switching entails steps of executing calling code, branching to a target code, executing linking code, branching from the operating system, and executing the target process.

OZAKI teaches the steps of: executing a portion of the domain code (application code for process T1), in the domain process context, calling for execution of targeted user code (selector call for T2_Proc); branching to a target code-segment corresponding to the targeted user code; executing linking-code in the target code-segment (call gate) and entering the operating system code in the domain process context (MB1); branching from the operating system code (MB2) in the user process context to the targeted user code (application code to process T2); executing the targeted user code (T2_Proc completed successfully); and returning to the domain code (T2 passes return parameters) (pg. 39). Refer to claim 1 for the motivation to combine.

As to claim 7, KARGER teaches a computer implemented method for multi-level memory domain protection by intra-group context switching between the user process context (calling domain context) and the domain process context (called domain context) and returning from such switching (col. 4, line 36 – col. 6, line 34). However, KARGER does not teach the context switching entails steps of executing operating

system code for returning; entering the target code-segment; executing linking code, returning to the domain code; and resume executing the domain code.

OZAKI teaches the step of returning comprises the steps of: executing the operating system code (MB2), in the user process context, calling for return to the domain code in the domain process context (return results); entering the target code-segment (RM1); executing the linking-code in the target code-segment to place the domain code (application code in T1) in a return state; returning to the domain code (data received in T1); and resume executing the domain code (pg. 39). Refer to claim 1 for the motivation to combine.

As to claims 4 and 5, refer to claims 6 and 7 for rejection. It would be obvious that since the context switching is performed from one calling domain to a called domain that either domain can function as either a user domain or a receiving domain.

As to claim 8, OZAKI teaches the user process context (T2) further includes user data (T2_Proc), and the steps of: executing a portion of the domain code (application code in T1), in a domain process context, calling for a data access from targeted user code (selector call for T2_Proc); accessing the user data located in the targeted user code (T2_Proc completed successfully); and resuming execution of the domain code ("When T1 receives the data from MB1 it continues executing.") (pg. 39).

As to claim 11, OZAKI teaches the steps of: executing calling-code (selector call for T2_Proc) in the first process pair (T1) calling for access to target data in the second process pair (T2); branching to a call gate target code-segment (call gate) in the calling-code corresponding to the target data; and branching to and accessing the target data (T2_Proc complete successfully and return parameters) (pg. 39).

As to claims 12-14, reference is made to a system which corresponds to the method of claims 1-3 and is therefore met by the rejection of claims 1-3 above.

As to claims 16-18, reference is made to a computer medium which corresponds to the method of claims 1-3 and is therefore met by the rejection of claims 1-3 above.

As to claim 21, OZAKI teaches the user code (application code of process T2) includes user-to-domain control transfer instructions (T2 passing returned parameters to T1), the system further comprising a user call gate (call gate), coupled to the user process and the domain process (pg. 39). It would be obvious that the call gate would store the user-to-domain control transfer instructions since it communicates with the mailboxes which interact with each other through the RMP.

As to claim 22, OZAKI teaches the domain code (application code for process T1) includes domain-to-user control transfer instructions (selector call for T2_Proc), the system further comprising a domain call gate (call gate), coupled to the user process

and the domain process (pg. 39). It would be obvious that the call gate would store the domain-to-user control transfer instructions since it communicates with the mailboxes which interact with each other through the RMP.

As to claim 23, KARGAR teaches the a plurality of programs that can call other domains (col. 6, lines 14-34). OZAKI teaches communication between two processes (pg. 39, Table 1). It would be obvious that the processor allows for communication between a plurality of processes and all communicate similarly to each other as disclosed in the independent claims.

As to claim 24, OZAKI teaches the user call gate (call gate) comprises a target code-segment for storing user-to-domain control transfer instructions (return parameters) which transfer control to a specific location in the domain code (application code in process T1) (pg. 39; "All mailboxes reside at privilege level zero and are accessed through call gates from lower privilege levels...If T2_Proc is completed successfully, then T2 passes the results to RMP via the mailbox. The results are then sent to T1. The RMP resets the timer once it receives a message from T2. When T1 receives the data from MB1 it continues executing.").

As to claim 25, KARGAR teaches the storing the stack frame and domain addres space location in order to switch contexts (col. 6, line 47 – col. 7, line 8). OZAKI teaches the target code-segment comprises: arguments to be passed from the user

process (T2) to the domain process (T1); and a data type description of the arguments (pg. 39, "If the call for T2_Proc is completed successfully, T2 passes return parameters to T1.").

As to claim 26, OZAKI teaches the target code-segment comprises linking-code, for handling how the user-to-domain control transfer instructions are executed (pg. 39, "T1 puts the time-out value and the selector for T2_Proc into a mailbox that it shares with RMP. Once RMP1 receives the message, it checkpoints the current state of T1 and sets an external timer that provides a hardware interrupt when a response is not received from T2_Proc in the specified amount of time.").

As to claim 27, OZAKI teaches the target code-segment comprises a calling-code return state, for storing a current state of the user process prior to when one of the user-to-domain instructions is executed (pg. 39, "..it checkpoints the current state of T1.."). It would be obvious that T2's state is also check pointed if T2 requires access of T1.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (703) 305-0439. The examiner can normally be reached on Monday-Friday, 8:30 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

lab